



Six paths from mainframe to Azure

While the mainframe remains a pillar of reliability, availability and serviceability, many customers are considering if their needs can also be covered in a cloud environment, such as Microsoft Azure. They should know that there are many roads to reach the cloud, and they can determine the pace in which they get there

Thinking about mainframe modernization paths is not new. Microsoft led the industry in mainframe migrations with the creation of the Mainframe Migration Alliance in April 2004. This alliance included an ecosystem of systems integrators and migration tool providers who delivered the services and tools to migrate from the IBM z/Series servers to the Windows Server platform. Customers wanting to reduce costs, increase agility and address an impending skills shortage embraced the Windows Server platform which delivered the Reliability, Availability and Serviceability (RAS) they required to run their mission critical workloads. Over the next decade, thousands of customers migrated (all or part of) their workloads from their mainframe servers to the Windows Server Platform.

With the Microsoft Azure cloud becoming widely accepted for its Reliability, Availability and Serviceability, there has been a new wave of customers considering mainframe migrations. The Microsoft Azure cloud is becoming the platform of choice for mainframe migrations.

A mainframe migration can be viewed as a journey with several stops along the way. Selecting the best method for performing a migration requires an understanding of the customer's business requirements and workloads. An analysis of each workload will determine the best strategy.

Options to be considered include:

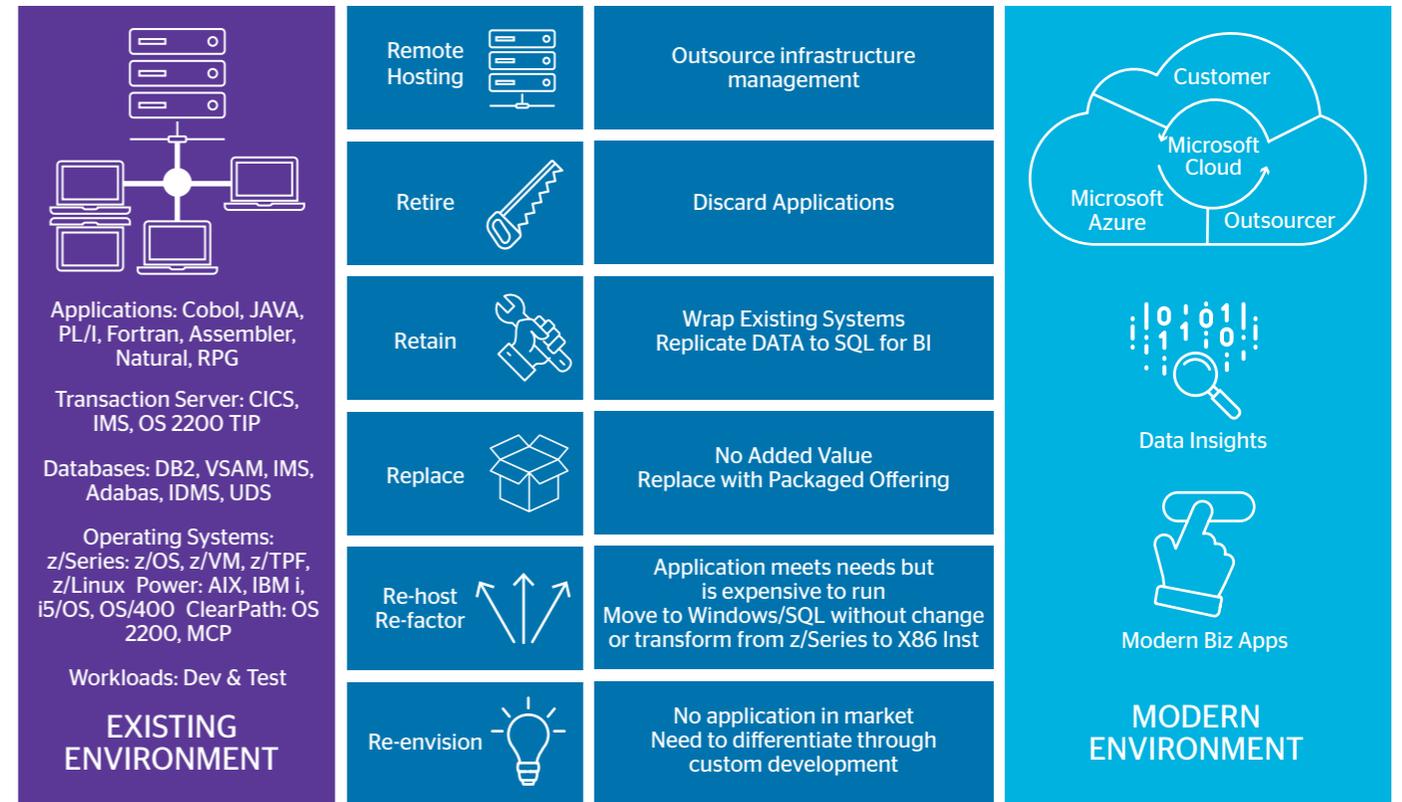
1. the remote hosting of the infrastructure to reduce cost and increase efficiency, the retirement of workloads that are no longer required
2. retaining workloads but accessing the data from a more agile platform
3. the replacement of workloads with a Commercial Off The Shelf (COTS) packaged applications
4. the re-hosting of workloads which is meeting the business requirements on a lower cost platform
5. the re-envisioning, re-architecting or re-writing of workloads which is not satisfying the needs of the business (see picture across page).

For some workloads and databases, the mainframe is an excellent platform which is delivering great business value. In those cases, the data can be accessed from a lower-cost, more agile platform which allows better access to the data for analysis and action. This can be accomplished by accessing the mainframe database or transactions through tools like the Microsoft Host Integration Services of IBM Websphere MQ. The mainframe database can also be replicated to SQL Server or the Azure SQL Database so that the data can be better analyzed through systems like R Server or the Cortana Intelligent Suite. This strategy would be categorized as a Retain strategy.

If a workload is meeting the needs of the business but is expensive to run on the mainframe, re-hosting or re-factoring to a lower cost platform is a good choice. Re-hosting and re-factoring technologies and options have improved significantly over the past decade. Re-hosting and re-factoring are preferred for some workloads as it can be performed at a lower cost and lower risk than Re-writing the workload.

When migrating a workload, the decision must be made whether it is best to continue maintaining the native language, to transform the source code to a new language or to extract the business logic from the source code and manage it in a vendor-provided Integrated Development Environment (IDE). If there are a sufficient number of programmers available to support the current language, re-hosting the original language but supporting it in a more modern IDE, like Visual Studio, is a good choice. If it is becoming difficult to find or retain a programming staff, then Re-factoring the source code to a new language would be a better choice. These choices are not mutually exclusive: once the programs are managed in Visual Studio or Eclipse, components can be re-written in C# or Java rather than staying in the original language. This strategy would be categorized as a Re-host/Re-factor strategy.

Regardless of your specific needs and environment, there is a modernization path available that will meet your exact requirements.



BUSINESS INITIATIVES DRIVE MODERNIZATION